

최소 개수의 스타이너 포인트를 위한 근사 비율 2 (Approximation ratio 2 for the Minimum Number of Steiner Points)

김 준 모 [†] 김 인 범 ^{**}
(Joonmo Kim) (Inbum Kim)

요약 본 논문은 STP-MSP을 위한 근사 알고리즘을 제안한다. 이 문제에 대해 근접한 최적 해법을 제공하는 PTAS를 가지는 것이 불가능하기 때문에, 본 논문의 연구는 $n^{O(1)}$ 의 실행 시간과 근사 비율 2를 가지는 하나의 대안을 제시한다. 본 연구의 중요성은 관련된 다른 미해결문제에 대하여 해결 가능성을 제시하는 것이다. 본 논문의 주요 제안내용은 문제 인스턴스에게 허용오차를 배분하는 것이다. 이로 인해 우리는 무한적 경우에서 다항적 범위로 실행시간을 줄일 수 있다. 관련연구[1,2]가 근사 비율이 2보다 크지만 보다 현실적인 실행시간을 갖는 근사 알고리즘들을 제시한 것이라면, 본 연구는 근사 비율이 2인 근사 알고리즘의 존재를 밝힌 것이다.

키워드 : 근사 알고리즘, 근사 비율, 허용오차, 다항적 범위, 실행시간, 문제 인스턴스, 동적 프로그램

Abstract This paper provides an approximation algorithm for STP-MSP(Steiner Tree Problem with Minimum number of Steiner Points). Because it seems to be impossible to have a PTAS(Polynomial Time Approximation Schemes), which gives the near optimal solutions, for the problem, the algorithm of this paper is an alternative that has the approximation ratio 2 with $n^{O(1)}$ run time. The importance of this paper is the potential to solve other related unsolved problems. The idea of this paper is to distribute the error allowance over the problem instance so that we may reduce the run time to polynomial bound out of infinitely many cases. There are earlier works[1,2] that show the approximations that have practical run times with the ratio of bigger than 2, but this paper shows the existence of a poly time approximation algorithm with the ratio 2.

Key words : Approximation algorithm, Approximation ratio, Error allowance, Polynomial bound, Run time, Problem instance, Dynamic program, STP-MSP(Steiner Tree Problem with Minimum Number of Steiner Points), PTAS(Polynomial Time Approximation Schemes)

1. 서론

본 논문은 PTAS에 관한 Arora의 연구 [3,4]에 기반을 둔다. 그의 대부분 연구가 최적화 문제들(optimization problems)에 대한 근접 최적화 해법(near-optimal solution)을 찾으려는데 반해, 본 연구는 근사(approximation)의 부차적인 목표를 위해 그의 개념을 이용할 수 있는 방법을 찾는 것이다. 그 부차적인 목표란 첫째는 근접 최적화 해법을 가질 수 없는 문제들에 대해 차선의 근사를 찾는 것이고, 둘째는 존재할 것으로 추정은

되지만 현재까지 발견하지 못한 알고리즘(algorithm)을 찾는 것이다. 그 개념은 알고리즘을 실행하는 각 단계에서 극소량의 error를 추가(adding)하는 것이지만, 본 논문의 제안은 error allowance의 양을 미리 할당(allot)하고, 그것을 지수적 복잡도(exponential complexity)를 다항적 복잡도(polynomial complexity)로 줄일 수 있는 기하학적 여백(geometrical margin)으로 변화시킨다. 결과적으로, 유연성(error allowance)을 가지는 쉬운 구조로 구체화된다.

본 논문에서는 PTAS 방법 [3,4]인 Rectangular Partition과 Dynamic Programming(DP)을 스타이너 트리(Steiner Tree) 문제 중의 하나인 STP-MSP(Steiner Tree Problem with Minimum Number of Steiner Point)에 적용한다.

[†] 정 회 원 : 한국정보보호진흥원 연구원
jmkim@kisa.or.kr

^{**} 정 회 원 : 김포대학 컴퓨터계열 교수
ibkim@kimpo.ac.kr

논문접수 : 2002년 11월 25일
심사완료 : 2003년 4월 17일

STP-MSP 문제정의 [1]: 유클리드(euclidean) 평면에서 n 개의 단말(terminal)과 양수 r 이 주어졌을 때, 각 에지(edge)의 유클리드 길이(Euclidean length)가 r 이하이고 모든 단말들을 최소 개수의 스타이너 포인트들 [5,6,7,8]로 연결하는 스타이너 트리(Steiner Tree)를 찾기.

이 문제는 VLSI 설계, WDM 광 네트워크(optical network), 무선 통신(wireless communication)등의 분야에 적용이 가능하며, 이것은 NP-hard 문제이다. 참고 문헌 [1]에서는 전통적인 유클리드 스타이너 트리 문제 [9]와의 차이와 그 응용에 대해서 기술하고 있다.

Lin과 Xue는 STP-MSP 문제는 NP-hard [10]이고 최소 스패닝 트리(Minimum Spanning tree)를 사용하는 근사가 근사 비율(approximation ratio) 5를 가진다는 것을 보였다. 참고문헌 [1]에는 개선된 방법의 근사가 정확히 근사 비율 4를 가진다는 것을 보였다. 또한 근사 비율이 3인 새로운 다항적 시간 근사와 어떤 특정 조건 하에서의 다항적-시간 근사 방법(Polynomial-Time Approximation Scheme)을 제시하였다. 최근에 보고된 이 비율은 2.5이다. 그러나 최적화 문제를 위한 그러한 기하학적(geometric) 분석들은 약간의 개선이 이루어질 때마다, 새로운 생각과 많은 노력이 필요로 할 것이다. 이에 덧붙여서 언급하고 싶은 것은 STP-MSP를 위해서, 직사각형 분할(rectangular partitions)과 일반적 동적 프로그래밍(dynamic programming)을 사용해서 $1+\epsilon$ 의 근사 비율을 제공하는, STP-MSP를 위한 PTAS의 디자인은 현재까지 성공하지 못했다는 것이다. 사실 그것은 불가능해 보인다.

본 논문의 연구는 근사 비율 2의 목표를 달성하는 것과 DP(dynamic programming)가 다항적 시간 내에 그것을 실행할 수 있음을 보이는 것이다. 비록 결과 근사 비율이 이상적인 $1+\epsilon$ 이지는 않더라도, 우리는 PTAS 방법을 사용한다. 결과적으로 STP-MSP에 대한 우리의 DP는 최적의 경우라고 추정되는 스타이너 포인트 개수의 최대 2배를 가지는 해법을 제시할 수 있다. 즉 결과적으로 최적화 비율 2를 얻는다.

2. 문제 인스턴스의 정의 및 변형(Definitions and the Modifications of the Problem instance)

본 논문은 STP-MSP에 대해서 관련 연구들 [3, 11,12]과는 다른 관점을 가지고 있다. 따라서 그것을 쉽게 기술하기 위해서 이 장에서는 기존의 존재하는 용어

[3]에 추가해서 새로운 용어를 정의한다. 첫 번째 사용하는 새로운 용어들은 밑줄 글씨로 표현하겠다.

근사에 대해서, 우리는 문제의 **loose version**을 위한 알고리즘을 디자인하려 한다: 사실, '스타이너 포인트들의 최소 개수'를 찾는 조건은, STP-MSP의 문제 정의에서 제거된다. 즉, 그 문제의 정의는 '유클리드 평면에 주어진 n 개의 단말과 양수 r 이 있을 때, 각 에지의 유클리드 길이가 r 이하가 되도록 스타이너 포인트들과 모든 단말들을 연결하는 스타이너 트리를 찾아라'이다.

그 다음에 우리는 문제의 해법으로 생성된 스타이너 포인트들의 수가, 스타이너 포인트들의 이상적인 최소개수의 두 배 이하인, 실행 가능한(feasible) 해를 제시할 수 있는 loose version의 알고리즘을 디자인할 수 있는 가능성을 보이고자 한다.

STP-MSP의 문제 인스턴스에 대해서, 우리가 loose version의 해가 되는 한 개의 스타이너 포인트의 집합을 알고 있다고 하면, 각 스타이너 포인트가 중심이 되는 원(circle)들을 그릴 수 있다. 평면 위에 존재하는 이러한 원들의 집합을 **Steiner-Cover**라고 부른다. **Steiner-Cover_{min}**는 스타이너 포인트들이 최적의 해(Optimal Solution)인 Steiner-Cover이다. 이것이 원래의 STP-MSP 문제의 해이다. 본 논문의 모든 원들은 상수 반지름 r 을 가진다. Steiner-Cover의 **size**는 그것에 포함된 원들의 개수이다.

본 논문에서는 다른 관련된 연구 [3,11,12]처럼 단말(terminal)에서 발단(portal)까지의 거리를 고려하는 것이 아니라, 단말이 원의 내부 혹은 외부에 위치하는 가를 체크한다. 우리는 단말을 격자(grid) 안으로 쉬프트(shift)하지 않는다. 왜냐하면 단말의 미세한 이동의 결과가 매우 큰 차이로 나타나기 때문이다. 격자의 단위 거리(unit distance)를 최소화하는 것은 문제 해결에 도움이 안된다. 그러나, 우리는 격자를 분석 목적으로는 사용할 것이다. 즉, 우리는 격자들을 따라서 선-분리자(line-separators)와 발단(portals)을 위치시켜 놓는다: 그들의 정의는 다음과 같다.

노드(node)들 집합의 **bounding-box**는 그들을 둘러싸는 가장 작은 직사각형 **rectangle**이다. **rectangle**은 bounding box의 한 분할(partition)인 축과 평행한 사각형(axis-aligned rectangle)이다. **rectangle**의 **size**는 **rectangle**의 긴 선분의 길이이다. **rectangle**의 **line-separator**는 **rectangle**의 짧은 에지와 평행한 직선이다. **line-separator**는 **rectangle**을 최소한 1/3의 면적을 가지는 두 개로 분리한다. 예를 들어, **rectangle**의 나비 W 가 그것의 높이보다 크다면, 어떤 **line-separator**는

rectangle의 가운데 $W/3$ 에 있는 수직인 임의의 직선이다. 이제 우리는 dynamic program이 수행되는 rectangle의 재귀적 분할(recursive partition)을 다음과 같이 정의한다.

Definition 1 ($1/3 : 2/3$ -tiling) : rectangle R 의 $1/3 : 2/3$ -tiling은 R 의 sub-rectangles의 이진 트리(hierarchy)이다. 이것의 root는 rectangle R 이다. R 의 size가 1 이하이면, hierarchy는 아무 것도 포함하지 않는다. 그렇지 않으면, root는 R 에 대한 line-separator와 line-separator가 R 을 분리해서 생성하는 두 개의 $1/3 : 2/3$ rectangle인 두 개의 서브트리(subtree)를 가진다.

타일링(tiling)을 통해서 깊이(depth)가 d 인 rectangles들은 root rectangle의 한 분할(partition)을 형성한다는 것을 주목해라. 깊이가 $d+1$ 인 모든 rectangle들의 집합은, size가 1보다 큰 깊이 d 의 각 rectangle을 관통하는 line-separator로 인해서 얻어진 분할이 세분화된 것이다. 깊이 d 인 rectangle의 최대 면적은 전체면적의 $(2/3)^d$ 이다. 따라서 다음과 같은 proposition을 도출할 수 있다.

Proposition 1 어떤 rectangle의 너비가 W , 높이가 H 이면, 그것의 모든 $1/3 : 2/3$ -tiling은 최대 깊이가 $\log_{1.5}W + \log_{1.5}H + 2$ 이다.

Definition 2 (Portals) $1/3 : 2/3$ -tiling에서 portal은 타일링(tiling) rectangle들의 둘레(perimeter) 위에 있는 포인트이다. m 이 양의 정수이고, 타일링의 각 rectangle의 line-separator 위에 정확히 m 개의 같은 거리(equidistant) portal들이 있다면, portal들의 집합 P 는 타일링(tiling)을 위한 m -regular라고 부른다(우리는 line separator의 끝점(end point)들도 portal이라 정한다. 즉, line-separator는 그것 위에 있는 portal들에 의해 정확하게 $m-1$ 개의 같은 크기로 분할된다).

points는 스타이너 포인트들과 주어진 단말들의 합집합(union)이다. 두 개의 포인트들 사이 혹은 rectangle의 한 변과 한 개의 point 사이의 거리가 r 이내라면, 이들은 **connectable**하다고 말한다. 어떤 rectangle내의 각 포인트들이 연결 가능한 쌍(connectable pair)을 가질 수 있다면, 이 포인트들의 집합을 **connectable**하다고 부르고, **connected**는 문제 인스턴스(problem instance)의 모든 포인트들이 연결 가능한 이웃 포인트(connectable neighboring point)를 가진다는 것을 의미한다.

PTAS의 관점에서, DP를 실행하기 위해서, 우리는 그것들이 line-separator들을 통과할 때마다 원들을

portal들과 교차(cross)시키는 것이 필요하다. 그러나 단지 몇 개의 원들만이 그렇게 될 수 있다. 실행성을 고려해서, 우리는 조건을 다음과 같이 변경한다. 한 개의 원과 한 개의 portal이 선택되었을 때, 그 원의 한 개의 포인트만이 portal과 교차해야 한다: 비록 같은 원의 많은 포인트들이 다른 portal들과 교차할 수 있어도, 우리는 그러한 교차를 중복(redundancies)으로 무시할 수 있다. 첨가해서, combination의 bounded number를 지키기 위해서, 우리는 각 원의 둘레를 따라 한 디바이스(device)를 놓는다. 즉 그림 1과 같이 원의 둘레를 따라 p 개의 등거리 포인트들을 표시하고, 그들에게 0부터 $p-1$ 까지의 색인을 붙인다. 그 후에, **indexed-point**들 중에서 단 하나에서 원과 portal이 만나도록 한다. 이러한 weak 조건의 타당성(feasibility)은 다음에 설명할 것이다. 서로 완전히 겹치는 많은 중복된 선택이 있을 수 있다. 그러나, 우리는 그들 중 단지 하나만을 무작위로 선택한다. 중복을 무시하는 것은 결과에 영향을 미치지 않기 때문이다.

Definition 3 (m -light Steiner-Cover) $m \in \mathbb{Z}^+$ 를 만족하는 m , bounding box의 $1/3 : 2/3$ -tiling을 S , 이 타일링 위의 portal들의 m -regular 집합을 P 라고 하자. 그러면 각 원들이 P 안의 최소 한 개의 portal들과 indexed points에서 만나는 Steiner-Cover는 S 의 관점에서 m -light하다.

같은 평면상에, 우리는 두 개의 Steiner-Cover들을 놓는다: 한 개는 다른 것의 복사본이고, 그 복사본의 각 원들은 약간 시프트(shift)된다. 그 후에, 우리는 한 쌍의 원을 선택하는데, 하나는 원본(original)이고 다른 것은 그것의 시프트된 복사본이다. 각 쌍에 대해, 중첩되지 않는 원의 영역을 **circle-distance**라고 부른다. **set distance**는 모든 쌍(pair)들의 모든 circle-distance들의 합이다.

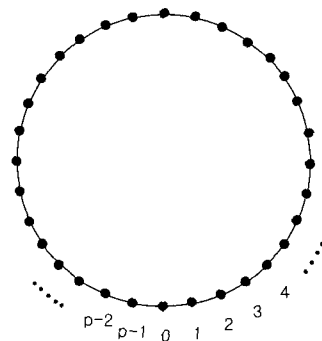


그림 1 Indexed-points

3. 근사 비율 2와 구조 정리(Approximation Ratio 2 and the Structure Theorem)

이 장은 근사 비율(approximation ratio)의 측정에 관한 내용이다. 우리는 원의 개수 관점에서 최소 크기의 *m-light Steiner-Cover*를 *Steiner-Cover_{opt}*와 비교한다.

첫째로, 우리의 목표가 왜 비율 2인가를 보인다. 우리가 제안하는 DP는 많은 가능한 경우에 실행가능성이 있는 해법들을 무작위로 시도하기 때문에, $1 + \epsilon$ 에 해당하는 원하는 비율에 대한 시도 수는 무한대(unbounded)이다. 그러나 다행히도 문제의 기하학적(geometric) 특성은 어느 정도 바람직한 비율 2를 얻을 수 있게 한다. 즉 우리는 그 비율을 위해서 다항적(polynomial)으로 가능한 경우의 수를 한정 할 수 있는 방법을 찾아 제안 할 수 있게 되었다.

본 논문의 연구는 STP-MSP 문제에 대한 최적 해법 트리(optimal solution tree)의 다음과 같은 3개의 특성 [1]을 토대로 하고 있다: 첫째 두 에지는 서로 교차하지 않고, 둘째 한 정점(vertex)에서 만나는 두 에지는 최소한 60° 의 각을 형성하고, 셋째 두 에지가 정확히 60° 의 각을 이루면, 그들의 길이는 같다.

*Steiner-Cover_{APX}*는 *m-light Steiner-Cover*중의 하나이며, *Steiner-Cover_{OPT}*로부터 ϵ 의 set-distance를 가진다. 그러나 우리는 DP가 *Steiner-Cover_{APX}*를 제한된 실행시간 동안 확실히 생성하는 것을 보장할 수 없음을 다음에 보이겠다. 따라서 우리는 또 다른 실행 가능한 해법(feasible solution)인 *Steiner-Cover_{2OPT}*를 목표로 삼는데, 이것은 *Steiner-Cover_{OPT}* 보다 2배 많은 원을 가진다. 우리는 DP가 생성할 수 있는 *Steiner-*

*Cover_{2OPT}*가 많이 존재함을 나중에 보인다.

원들이 단말(terminal)들을 cover해야 한다는 요구조건을 잠시 생략하면, set-distance의 관점에서 *Steiner-Cover_{APX}*를 DP에 의해 얻을 수 있다. 그러나 원래의 *Steiner-Cover_{APX}* 조건으로 돌아가면, set-distance의 어떠한 작은 영역 안에도 cover되지 않는 단말(uncoverd terminal)들이 존재할 수 있고, DP는 그것들을 cover하기 위해 더 많은 원을 만들게 되므로 *Steiner-Cover_{APX}*는 DP에 의해 얻을 수 있으리라는 확신을 할 수 없다. 그러한 경우는 그림 2에서와 같이, *Steiner-Cover_{opt}* 원들의 둘레 바로 위에 단말들이 존재할 때, 확실히 나타난다.

두 중심의 거리가 r 이내에 있는 두 개의 *covering-circles*가 한 개의 *given-circle*을 부분적으로 cover하는 경우를 고려해보자. 이 경우를 위해서 다음과 같은 정의가 가능하다. covering-circle들의 중심이 결국 스타이너 포인트이기 때문에, 그들은 거리 r 이내에 위치해야 한다: 그림 4의 음영으로 표시된 내부의 작은 원은 두 covering-circle들의 중심들이 위치할 수 있는 영역을 표시한다.

한 원에 대하여, 둘레의 선들을 일정한 너비(width) w 를 갖도록 그리면 도넛(doughnut)모양을 얻게 된다. 우리는 이것을 *circular-band*라고 부른다. 두 개의 covering-circle들에 대해서, 이 원들이 circle-band들에 의해 치환된다면, 우리는 그것을 *c-band*라고 부른다(그림 3). 그림 4와 같이, c-band의 너비 w 는 다음과 같은 조건을 만족해야 한다: (1) 포인트 a 에 가장 가까운 c-band의 내부 둘레(inner-perimeter)는 a 위(above)에 놓여져야 한다. (2) b 에 대해서도 같은 방식으로 적용된다. (3) 포인트 c 의 가장 가까운 내부 둘레의 교차점은

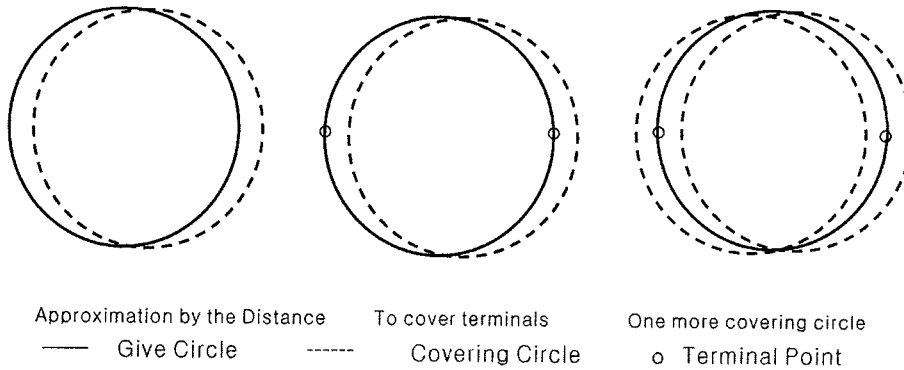


그림 2 Difficulty for approximation by ϵ movements

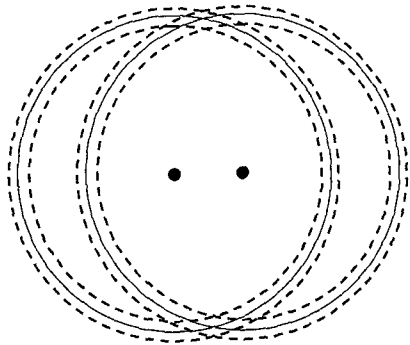


그림 3 c-band

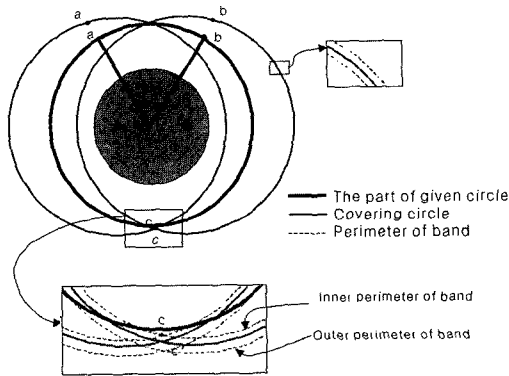


그림 4 Covering with enough margins

c 아래(bellow)에 놓인다.

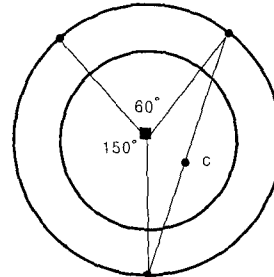
' c -band 내부의 원들의 한 쌍'의 의미는 c -band의 두 개의 circular-band 각각이 내부에 한 개의 circle을 포함하는 경우를 의미한다. inner-area는 두 개의 내부 둘레의 안에 있는 두 개 영역의 합집합(union)이다.

Lemma 1 (m -cover: covering with a margin) 중심이 최단 최적 해법 트리(shortest optimal solution tree)의 스타이너 포인트인 어떤 원과 그 위의 단말들에 대하여, inner area가 모든 주어진 단말들을 cover하는 c band가 존재한다.

증명 :

이 상의 앞에서 언급했던 특성들[1]과 원의 중심이 최적 경우의 스타이너 포인트라는 사실에 의해, 임의의 이웃하: 두 예지 사이의 각도는 반드시 60° 보다 커야한다. 여기서 예지란 스타이너 포인트와 한 단말(terminal) 사이의 선이다.

그림 5와 같이 150° , 150° , 60° 의 3개의 부채꼴(fan)이 생성되도록 세 개의 포인트를 원 둘레 위에 위치시킨



The diameter of the small circle and the radius of the big circles are length r
 ● vertex ■ center of a circle

그림 5 Covering : Analytic view

다. 150° 각을 가지는 큰 부채꼴의 둘레 상에 위치한 두 개의 정점(vertex) 사이에 한 개의 선을 그리고, 그 선의 중심을 c 라 한다. c 를 중심으로 해서 큰 부채꼴을 cover하는 새로운 원을 그릴 수 있는데, 그 이유는 c 로부터 처음 원의 중심까지, 각 정점(vertex)들까지, 부채꼴의 호(arc)까지의 거리가 모두 r 미만이기 때문이다. 같은 방법으로 이것을 다른 큰 부채꼴에 적용할 수 있다. c 와 다른 큰 부채꼴에 있는 또 다른 c 사이의 거리는 $2r \cdot \cos 75^\circ \cdot \cos 15^\circ = 1/2 \cdot r$ 의 계산으로 얻을 수 있다. 그러므로 Lemma 1의 c -band는 두 개의 새로운 원들을 기반으로 생성될 수 있다. ■

Theorem 1 (Structure Theorem) R^2 상에서의 단말들의 집합은, 그 집합의 Steiner-Cover_{OPT}를 m -cover하는 무한개의 Steiner Covers_{2*OPT}를 가지며, 그 중 일부는 DP를 이용해서 계산해 낼 수 있다. 이 집합 위에 bounding-box의 1/3:2/3-tiling을 설정할 수 있으며, Steiner-Cover_{2*OPT}는 이 타일링 상의 m -light이다. 여기서 $m=O(\log n)$ 이고 L 은 bounding-box의 size이다 (그림 6 참조).

증명 :

Lemma1(m -cover)에 따라서, 임의의 Steiner-Cover_{OPT}는 다음과 같은 Steiner Covers_{2*OPT}를 가질 수 있다. 위 두 개의 Cover를 같은 평면상에 위치시킬 때, Steiner Covers_{2*OPT}내의 원(circle)들의 각각의 쌍이 Steiner-Cover_{OPT}의 각각의 원 C_{opt} 을 m -cover 한다. 그 쌍을 pair_{2*opt}라고 이름 붙인다. 한 개의 pair_{2*opt}는 c -band를 가지는데 이 band에는 pair_{skin}이라고 이름 붙인 각 circle들의 무한개의 pair들의 집합을 포함한다(그림 7). 각 pair_{skin}은 inner-area를 포함하기 때문에 C_{opt} 상의 모든 단말들을 cover한다. 따라서 이것은 기능적으로 pair_{2*opt}와 동일하다. 우리는 pair_{2*opt}를 pair_{skin} 중의 하나라고 볼 수 있다. 이것으로 모든 원들이

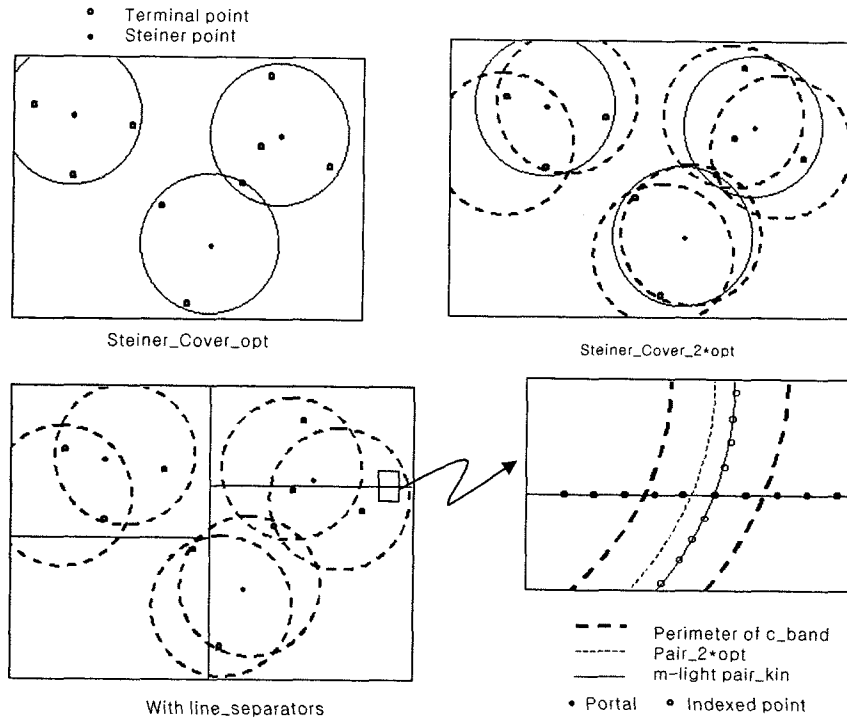


그림 6 m-light Steiner-Covers_{skin}

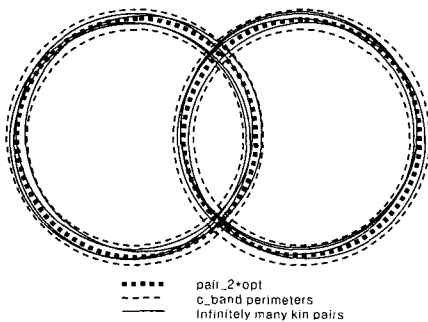


그림 7 Pairkin

pairs_{kin}들 중의 하나인 m-light Steiner-Cover가 존재한다는 사실을 유도할 수 있다. 이제 우리는 Steiner-Covers_{skin}을, 각 m-covering pair가 하나의 pair_{kin}인 Steiner-Covers_{2*OPT}의 일반형으로 정의한다.

DP가 실행되고 타일링(tiling)이 구성될 때, Steiner-Covers_{2*OPT}의 각 c-band 영역은 line-separator에 의해 교차된다. c-band의 너비가 상수 w 이기 때문에, c-band 내부에 놓여있는 line-separators를 따라 portal들이 존재하도록, w 보다 작은 d 를 inter-portal dis-

tance로 선택해야 할 것이다.

또한 Steiner-Covers_{skin}이면서 m-light Steiner Covers인 것이 존재할 수 있도록, 색인된 포인트들의 수 p 를 충분히 크게 설정할 수 있다. Steiner-Covers_{skin}은 같은 평면상의 가상의 Steiner-Cover_{OPT}를 근거로 해서 생각될 수 있다. 즉, m-light Steiner-Covers_{skin}이 존재할 수 있는데, 이것은 m-light Steiner-Covers이면서 동시에 Steiner-Cover_{2*OPT}의 친척(kin)이다. Steiner-Cover_{2*OPT}의 친척은 Steiner-Cover_{2*OPT}처럼 Steiner-Cover_{OPT}를 cover한다. 다음 장에서는 m-light Steiner-Covers_{skin}이 DP에 구성될 수 있음을 보인다. ■

Steiner-Cover_{OPT} 내의 원들은 아래 기술된 내용에 따라 5회 이상 중복되지 않는다. 그 line-separator와 원의 둘레와의 교차회수는 상수에 의해 제한된다. 그래서 이 문제에 있어서, 교차 회수가 m 이상인 경우는 고려할 필요가 없다.

교차회수는 다음 세 가지 요인에 의해 상한이 결정된다. 첫째 평면상에서의 단위 길이 r 의 격자(grid)에 대해, 우리는 모든 격자의 포인트들을 원의 중심으로 해서 원들을 그릴 수 있다. 그러면 원들은 격자의 모든 영역을 완전히 cover한다. 그러므로 모든 단위 영역

(unit area)을 cover하는 데는 5회를 초과한 중복이 필요하지 않으므로 Steiner-Cover_{OPT}는 모든 단위 영역을 cover하기 위해 5회를 초과한 중복은 필요 없다. 그림 8을 살펴보면 길이 $2r$ 이상의 line-separator의 교차회수는 상수 8회로 제한됨을 알 수 있다. 그림 8에서 격자의 가운데에서의 교차회수는 10회처럼 보이나 2회의 교차는 다른 부분에서 수행된 것이다. 둘째는 bounding box size인 상수 L 이고 셋째는 원들의 반지름인 상수 r 이다.

그러므로, 교차의 최대회수는 $\frac{8L}{r}$ 로 표기할 수 있다. 참고문헌 [3,11,12]의 관련 연구와는 다르게 우리는 교차회수가 m 보다 큰 경우를 무시할 수 있다. 그 이유는 우리는 $\frac{8L}{r}$ 을 초과되는 m 을 선택할 수 있고, m 을 $m = O(\log n)$ 으로 설정할 수 있기 때문이다.

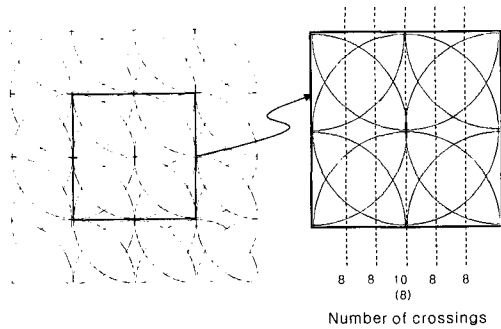


그림 8 Complete cover and max crossing

4. 동적 프로그래밍(The Dynamic Programming, DP)

Structure Theorem은 m -light Steiner-Cover_{kin}과 그와 연관된, $m = O(\log n)$ 인 타일링 S 가 존재함을 보장한다. 따라서 최소 크기의 m -light Steiner-Covers를 찾는 일은 비율 2의 근사(approximation)를 구하는 것이다. 비율 2는 최악의 경우를 가정했을 때의 결과임을 유의해라. proposition 1에 의해, 그러한 타일링의 깊이(depth)는 최대 $O(\log n)$ 이다. 이 장에서는 S 와 m -light Steiner Cover_{kin}을 다항적 시간 $n^{O(1)}$ 이내에서 찾는 DP에 대해 설명한다.

일반적으로 DP의 기능은 많은 선택들의 조합을 다항적으로 계산함과 동시에 각 선택에 대해 최소비용 구조(연결 그래프)를 구성하는 것이다. 어떤 rectangle에 대해서, portal들과 색인된 포인트들의 선택에 따라 그것의 원들이 결정되고, DP는 원들의 중심(center)들이, top rectangle 위치에서의 problem을 해결할 수 있는

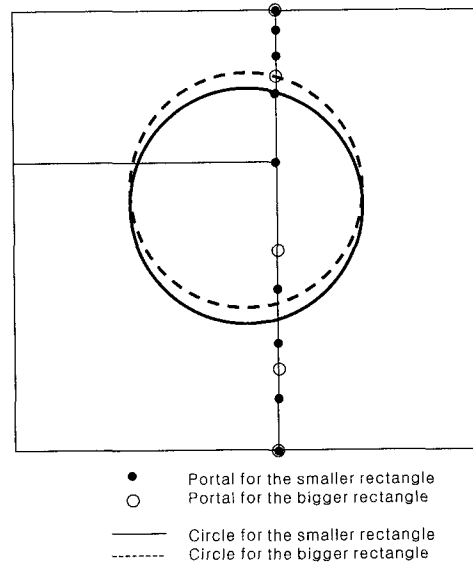


그림 9 Moving a circle

스타이너 포인트(Steiner point)들이 될 수 있는 여부를 체크한다. 연결성(connectivity)에 관점에서, rectangle 내부의 모든 포인트들이 connectable 하다면, 상위 레벨의 rectangle에서의 다음 작업을 위해 accept될 수 있을 것이다. 복잡도(complexity) 관점에서, 원의 중심(Steiner point)이 필수적인 연결(connection)을 만들지 않고 중복적인 것을 만든다면, 그 원은 다음 작업에서 제거될 수 있다.

두 개의 작은 rectangle이 큰 rectangle을 만들기 위해 결합될 때, 작은 rectangle의 portal 위치는 큰 것의 portal들과 정확히 맞추어지지 않을 것이다. 그래서 우리는 그림 9와 같이 작은 rectangle의 portal을 큰 rectangle의 가장 가까운 portal로, 원과 함께 이동시킨다. c-band의 너비는 이동을 위해 약간의 freedom margin을 주기 때문에, 이 움직임은 허용 가능하다. 우리는 가장 위쪽 rectangle의 line-separator위에 있는 두 개의 이웃하는 portal의 거리를 조절할 수 있다. 그리고 이 거리가 c-band의 너비보다 충분히 작다면, 작은 움직임은 허용 가능하다. 이것은 참고문헌 [3,11,12]의 연구에서의 bridges와 유사하다.

rectangle 조합의 각 경우는 DP의 lookup 테이블내의 한 개의 entry를 가지고 있고, 그것은 최소 개수의 circle들(Steiner Points)로 connectable 포인트 집합을 구성하는 더 작은 rectangle들의 entry들을 가리킬 것이다. 각 경우에 사용되는 circle들의 최소 개수는 테이블

entry에 적절히 기록된다. 실제적으로 우리는 평면상의 무작위 원 할당(random circle allocation)에 관한 다항적인 많은 경우(polynomial many case)를 모두 체크한다. 경우들의 수를 다항적으로 유지하기 위해서, 우리는 큰 rectangle이 형성될 때 작은 rectangle 내부의 원들을 이동시킴으로 허용 오차(error allowance)를 이용했다. 결국 우리는 각 단계에서 두 개의 rectangle을 합해 하나의 큰 rectangle을 형성하면서도, 각 rectangle 내부에서의 조합 경우(m portals에서의 선택)의 수를 같은 값이 되도록 유지하였다, 즉, 경우의 수를 증가시키지 않음으로써 복잡도의 지수적 증가를 막았다.

DP의 마지막 단계에서, 최상 위 rectangle인 *bounding-box* 내에서 원 배치(circle allocation)의 각 경우에 대해서, 우리는 그것이 Steiner-Cover인지를 체크하고, 그 rectangle의 모든 테이블 entry들을 비교하여 최소의 원 수를 가지는 것을 찾는다. 발견된 것이 원하는 최종 결과이다. 문제가 NP-hard이므로, 어떤 주어진 원의 배치가 그 문제의 해인지의 여부를 체크하는 것은 다항적 시간 내에 종료된다.

이제 우리는 DP 작업을 분석적 방법 관점에서 본다. DP 작업형태는 상향(bottom-up) 접근이다. 그러나 그 프로시저를 최종 단계에서 시작단계, 즉 하향(top-down) 방식으로 보는 것이 쉽다. DP의 최종 결과물은 rectangle로서, 이것은 주어진 point set을 위한 *bounding box*이고, 또한 이것은 문제를 비율 2로 해결하는 *m-light structure*를 포함한다.

우리는 문제 인스턴스를 포함하는 bounding box를, 1/3:2/3-tiling을 반복적으로 적용해서 분할시킨다. 이것은 brute-force algorithm을 사용해서 각 rectangle 내에서 포인트들의 *connectable set*을 다항적 시간 내에 찾을 수 있도록, 모든 최하위(bottom most) rectangle들이 작은 수의 단말을 포함할 때까지 반복된다. 즉, 각 최하위 rectangle들에 대해서, 우리는 line-separators 둘레 portal 선택의 가능한 모든 경우인 2^{4m} 개와 어떠한 원 둘레의 색인된 포인트 p 개의 선택을 고려한다. 각 경우의 조합에 대해, DP는 해당 원들을 그리고, 새로이 생성된 포인트들의 *connectable* 여부를 점검한다.

상위 단계의 rectangle에서, *line-separators*는 1/3 : 2/3-tiling으로 결정된다. 상위 단계의 rectangle을 위해 다음 3가지 요인들의 조합에서 연결 가능한 포인트 집합을 찾아야 한다: 첫째는 rectangle 내부의 *line-separators*이고 둘째는 그것을 따라 있는 *portals*, 셋째는 원의 색인된 포인트들의 선택이다.

rectangle에서 조합들의 각 경우에 대해, 우리는 왼쪽과 오른쪽 sub-rectangles에 대한 포인트들의 *connectable set*을 이미 계산하였다. 우리는 현재 rectangle의 연결 가능한 포인트들의 집합을 얻기 위해, 두 개의 sub-rectangle에서 얻은 두 개의 포인트들 집합들을 연결한다. 이 방법으로, 타일링의 root까지 가능한 모든 경우의 포인트들의 *connectable set*을 찾는 것이 가능하다: root에서, DP는 최종결과로 스타이너 포인트들(혹은 원들)의 최소 개수를 가지는 포인트들의 *connected set*을 선택하게 된다.

이제 우리는 이 DP를 위한 lookup 테이블의 entries 개수가 다항적(polynomial)이고 각 entries를 위해 소요되는 실행시간이 다항적임을 보인다. 한 개의 entry는 다음과 같이 3중(triple)으로 색인 된다: (a) 한 개의 rectangle (b) rectangle 둘레에 있는 $k_1(\leq 4m)$ portals의 집합 (c) k_1 개의 원 위치의 선택, 즉 p 개의 indexed-points $\{0,1,2,\dots,p-1\}$ 로부터의 순열 크기(permutation size) k_1 .

먼저 (a)에 대해서 설명하면, 각각 다른 rectangle들의 수는 최대 $\binom{n}{4}$ 이다. (b)에 대해서는, 각 rectangle은 상위 단계의 어떤 rectangle의 *line-separator*들의 일부인 4개의 *side*들을 가진다. *line-separator* 상의 m portal들은 같은 간격으로 떨어져 있고, 따라서 그것들은 우리가 알고있는 *line-separator*들로 한번에 완전히 결정된다. *line-separator*의 선택 회수가 최대인 경우는 포인트들의 pair 수, 즉 $\binom{n}{2}$ 이다. 따라서 이것은 $O((n^2)^4) = O(n^8)$ 을 설명한다. 더욱이 우리가 4개 side 위에 있는 $4m$ 개 이하의 portal들의 집합을 알고있다면, k_1 portal들의 집합(set)을 선택하는 방법의 수는 $\binom{4m}{k_1}$ 이다. 그래서 (b)에서의 선택들은 $n^8 \times \sum_{k=1}^c \binom{4m}{k}$ 이고, c 는 교차의 최대수인 $4 \cdot \frac{8L}{r}$ 이다. (c)에 대해서는, 위에서 선택된 각 portal에 대해서, 원 모양의 p 개의 선택이 있다. 그러므로 우리는 lookup 테이블의 크기를 다음과 같은 상한으로 정리할 수 있다.

$$n^4 \times n^8 \times \sum_{k=1}^c \binom{4m}{k} p^k = O(n^{12} \times (2p)^{4m}) = n^{(11)}$$

여기서 $m=O(\log n)$, p 는 상수, c 는 $4 \cdot \frac{8L}{r}$ 이다.

다음으로 lookup 테이블에서의 실행시간을 고려해보자. 맨 아래 단계의 rectangle에서, 원을 그리고, 선택된 것의 연결성 여부를 체크하는 시간은 상수이다. 왜냐하면

우리는 rectangle들이 정해진 수의 단말들을 갖도록 설정했기 때문이다. 상위 단계(upper level)의 rectangle들에서, DP는 두 개의 sub-rectangle들의 비교하고, line-separator들을 따라 제한된 이동을 시키고, 두 개를 접합시키며 포인트들의 연결성 여부를 체크함으로써, 결과적으로 connectable 집합을 구축한다. 위의 실행시간 대부분은 연결성 검사에 할당되고 그 밖의 다른 작업들은 모두 상수항이므로, 상위 단계 rectangle을 위한 실행시간은 최대 n^2 이다. 그러므로 DP의 실행시간은 $n^2 \times \text{태이블의 크기}$ 로 상한 되어지고, 이것도 역시 $n^{O(1)}$ 이다.

결론적으로, DP는 모든 *m-light Steiner-Cover*를 체크하게 되는데, 그 이유는 어떤 rectangle내에서도 그 존재가 Structure Theorem에 의해 보장되는, *m-light Steiner Cover_{kin}*를 최악의 경우이라도 찾아내기 위해서이다.

5. 결론

결과적으로, 본 논문의 전체적인 내용은 최적(optimal)은 아닌 실행 가능(feasible)한 다른 목표에 도달하기 위한 근사(approximate)기법에 관한 것이다. 이 목표치는 최적치의 두 배이므로 그 비율은 $1 + \epsilon$ 대신 2가 되지만, 결국 우리는 DP에 의해 다항적 시간(poly-nomial time)내에 그런 결과를 얻을 수 있게 되었다.

DP가 실제 문제(real problems)에 적용된다면, 단말들은 수학적인 포인트들이 아니고, 작은 영역을 각각 가질 것이다. 그래서 2보다 작은 비율을 가지는 거의 최적의 해를 얻는 것이 가능할 것이다. 그러한 실제적인 목적을 위해 사용하려면, c-band의 정의만 변경 해주면 될 것이다.

이 문제의 중요성은 'Minimum Disc Cover' 문제와 비슷하거나 그 이상이다. 그와 관련된 문제들은 본 논문의 방법을 조금 수정하면 해결할 수 있다. 여기서 강조해야 할 사실은, 이 방법은 관련된 다른 문제들에 대해 일관성 있는 해결안을 제시한다는 것이다. 각각의 문제를 기하학적 분석(geometric analysis)으로써 해결하려는 경우에는, 일반적으로 문제별로 서로 방향이 다른 연구를 해야한다.

한편으로, 우리는 비율 2 이하의 STP MSP 근사를 위해서는 어떤 실마리도 찾지 못하고 있다. 더욱이 목표값(objective value)이 정수라는 사실은 PTAS를 가지는 것이 불가능하다는 하나의 표시가 될 것이다.

참고 문헌

[1] Donghui Chen, Ding-Zhu Du, Xiao-Dong Hu, Guo-Hui Lin, Lusheng Wang and Guoliang Xue, Approximations for Steiner Trees with Minimum Number of Steiner Points, *Journal of Global Optimization*, Vol. 18, No.1, pp. 17-33, 2000.

[2] L. Wang and D.-Z. Du, Approximations for a Bottleneck Steiner Tree problem, *Algorithmica*, Vol. 32, pp. 554-561, 2002.

[3] S. Arora, Polynomial time approximation schemes for Euclidean TSP and other geometric problems, *Proc. 37th IEEE Symp. on Foundations of Computer Science*, pp. 2 12, 1996.

[4] S. Arora, Nearly linear time approximation schemes for Euclidean TSP and other geometric problems, *Proc. 38th IEEE Symp. on Foundations of Computer Science*, pp. 554-563, 1997.

[5] F.K. Hwang, D.S. Richards and P.Winter. The Steiner Tree Problem, *Annals of Discrete Mathematics*, Vol.53, North-Holland, 1992.

[6] M.R. Garey, R.L.Graham and D.S.Johnson, The complexity of computing Steiner Minimal Trees, *Journal on Applied Mathematics*, Vol. 32, pp. 853-859, 1997.

[7] E.N. Gilbert and H.O.Pollak, Steiner minimal trees, *SIAM Journal on Applied Mathematics*, Vol.16, pp. 1-29, 1968.

[8] Ding Zhu Du, F.K. Hwang. A Proof of Gilbert Pollark's conjecture on the Steiner ratio, *Algorithmica*, Vol. 7, pp. 121-135, 1992.

[9] D.Z. Du and F.K. Hwang, An approach for proving lower bounds: solution of Gilbert Pollak conjecture on Steiner ratio, *Proceedings of IEEE 31st FOCS*, pp. 76-85, 1990.

[10] G. H. Lin and G.L.Xue, Steiner tree problem with minimum number of Steiner points and bounded edge length, *Information Processing Letters*, Vol. 69, pp. 53-57, 1999.

[11] X.Cheng, J. M. Kim and B. Lu, A Polynomial Time Approximation Scheme for the Problem of Interconnecting Highway, *Journal of Combinatorial Optimization*, vol.5, No.3, pp. 327-343, Feb. 2001.

[12] J. M. Kim, X.Cheng and Ding Zhu Du, A PTAS for Grade of Service Minimum Tree Problem, Technical Report of University of Minnesota.



김 준 모

1989년 서울대학교 컴퓨터공학과(공학사). 1990년~1993년 공군 중앙전산소
1993년~1994년 DACOM. 1996년~1997
년 Mathematics, University of Toron-
to(학부과정). 2001년 Computer Science,
University of Minnesota(공학박사).
2002년~현재 한국정보보호진흥원 선임연구원. 관심분야는
Approximations for NP-hard problems, Church's
Thesis, 실행프로그램 분석기술 등



김 인 범

1989년 서울대학교 컴퓨터공학과(공학사). 1991년 서울대학교 컴퓨터공학과
(공학석사). 1991년 2월~1994년 7월 대
우통신 종합연구소 연구원. 1995년 1
월~1996년 2월 오라클 코리아 연구개발
실 근무. 1996년 3월~현재 김포대학 컴
퓨터계열 조교수. 관심분야는 컴퓨터이론 및 알고리즘, 데이
타베이스 시스템, 컴퓨터구조 및 시스템, 컴퓨터보안 등